

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
1. Abend	<p>RaspberryPi mit Sense-Hat in Betrieb nehmen und erstes «Hello.py» zur Ausführung bringen.</p> <p>Ablaufstrukturen im Programm gezielt und richtig verwenden, <i>print()</i> und <i>input()</i> Funktionen sicher anwenden.</p> <p>Umrechner.py (ohne eigene functions und ohne Bildschirmsteuerung)</p>	<p>Vorstellung (Wer bin ich? Problem-Based Learning)</p> <p>Installieren der Entwicklungsumgebung</p> <ul style="list-style-type: none"> • VNC / Notepad++ / PuTTY / FTP Client • Nano auf RPi, die wichtigsten Befehle • 1.Programm <ul style="list-style-type: none"> ◦ File erstellen und editieren (Console, Nano) ◦ <code>print()</code>, <code>#!/usr/bin/python3</code> ◦ LINUX Befehle: <ul style="list-style-type: none"> ▪ <code>cd</code>, <code>ls -al</code>, <code>pwd</code>, <code>rm</code>, <code>mv</code>, <code>cp</code>, <code>mkdir</code>, ▪ LINUX file-system (<code>chmod</code>, <code>filepath</code>) ◦ Execution ◦ Fehlermeldungen interpretieren können und Lösungen implementieren • Notepad++, Putty • Programm erweitern (<code>Print()</code>, String-Operationen, <code>Input()</code>) <p>Aufgabe 1a (Umrechner.py)</p> <ul style="list-style-type: none"> • Menu • User-Input (Wähle:) • If-then-elif-else Struktur • Loop mit 0 beenden • Behandlung von falsch Eingaben • Formeln implementieren (Variablen, Float-Input, Math-Operationen) • <code>format()</code> Methode 	<p>Nach Anleitung installieren und konfigurieren</p> <p>Selber versuchen, Vormachen, Nachmachen mit theoretischen kurzen Einschüben</p>	<p>10'</p> <p>60'</p> <p>130'</p>	<p>Umrechner.py alle Formeln implementieren und alle Menu-Punkte vollständig implementieren.</p>

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
2. Abend	<p>Eigenen Funktion unter dem Aspekt der Re-Usability implementieren können.</p> <p>Den Unterschied zwischen positional und named Parameters bei den Functions-Interfaces wie beim Aufruf sicher und gezielt anwenden können.</p> <p>Exception-Handling in Python sicher anwenden und den Unterschied zwischen pre-condition check and exception sicher anwenden können.</p>	<p>Aufgabe 1b (Umrechner.py)</p> <ul style="list-style-type: none"> Bildschirmsteuerung (cls(), halt()) implementieren import math Formeln in Funktionen implementieren Funktionen abwärtskompatible erweitern Exception Handling mit Pre-Checks und try-catch <p>Aufgabe 1c (Umrechner.py, xxLibrary.py)</p> <ul style="list-style-type: none"> Funktion in eigene Library auslagern Refactoring Umrechner.py verwendet eigene Library Weitere Funktionen readInt(), readFloat() implementieren, testen, anwenden und in eigene Library übernehmen. Neuer Menu-Punkt: Quadratische Gleichung 	<p>Test-Driven Approach mit theoretischen Einschüben</p>	<p>100'</p> <p>100'</p>	<p>Neue Funktionen entwickeln, testen und in eigene Lib übernehmen.</p>

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
3. Abend	<p>LINUX Basics</p> <p>Alle Methoden im SenseHat Module (gemäss API doc) erfolgreich selbst getestet.</p> <p>Methoden Sense-Hat und Sense Klasse (API) mit LED Matrix verwenden.</p>	<p>Leistungskontrolle 1</p> <p>Aufgabe 2a (LED_Matrix.py)</p> <ol style="list-style-type: none"> 1. setPixel(), setPixels(), clear(), sleep(), showMessage() 2. Eventhandling (Joystick) 3. IMU- und Meteo-Sensoren <p>Aufgabe 2b (xx_SenseHat_Librarie.py)</p> <ol style="list-style-type: none"> 1. setPixel() mit clipping 2. drawLine(), drawRecantgle(), drawCircle() 3. Functions erweitern mit fillColor und borderColor 4. drawCompassNeedle(azimutInGrad) 	<p>Moodle Test</p> <p>Test-Driven Approach mit theoretischen Einschüben</p> <p>Test-Driven Approach mit theoretischen Einschüben</p>	<p>15'</p> <p>60'</p> <p>140'</p>	<p>Design und Implementation eines analogen Kompasses (mit Nadel)</p>

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
4. Abend	Containers in Python kennen und in eigenen Applikationen anwenden können.	<p>Elemente in den verschiedenen Containers zugreifen (lesen), zufügen/ändern und löschen. Listen[], Tupels(), Dictionaries{}</p> <p>Sub-Listen mit [1:-1] ranges lesen resp verarbeiten/ändern.</p> <p>for – Loops</p> <ul style="list-style-type: none"> • Listen und Tuples • Dictionaries (keys()) • <p>Comprehensions mit Filter und ZIP für eigene Anwendungen einsetzen können.</p>	Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben	90' 110'	<p>Meteo-App oder einer Snake-App oder</p> <p>Linien Aufgaben</p>

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
5. Abend	Containers Algorithmen in Funktionen umsetzen Parameterübergaben * (listen) ** (dictionaries)	Leistungskontrolle 2 <ul style="list-style-type: none">• Fakultät• Primzahlen Rechner• Primzahlen und Teiler Listen• Filter-Berechnungen	Formativer Test Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben	40' 160'	

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
6. Abend	REST-Service mit JSON Response nutzen	Open-Weather REST Service mit eigenem Token (AppID) aus Python aufrufen (requesten) und response als JSON Struktur verarbeiten.	Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben	40'	Design und Implementation einer Meteo-Logger (Wetterstation), welche Meteo-Daten von einem Ort / Lokation optimiert und ohne «Löcher» loggen.
	Filehandling und direct EXCEL Zugriff erfolgreich anwenden	Filehandling open() for read, write and append (inkl UTF and ASCII)		40'	
				120'	

Unterrichtsplan

Walter Rothlin

7. Abend	Klassenkonzept in Python in einer konkreten Anwendung kennen lernen und anwenden können.	<p>Eine eigene, allgemein einsetzbare Logger-Klasse gemäss Spezifikation entwickeln und testen.</p> <p>Anschliessend eigene Logger-Klasse in Meteo-App einsetzen.</p>	Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben	200'	
8. Abend	<p>Multi-Treathing und Timer-Events in Python kennen lernen.</p> <p>Eine Wrapper-Class für einen Wetterdienst allgemein und nach OO Ansätzen designen und implementieren.</p>	<p><u>Leistungsnachweis (Modullernzielkontrolle MILZ):</u> Eine allgemeine Weather-Class designen und implementieren, welche eine Wetterstation an einem bestimmten Ort kapselt.</p>	<p>Konzept anhand einiger Beispiele erklären (Walk-Through)</p> <p>Selbstständiges programmieren und individuelle Reviews durch Dozent.</p>	<p>10'</p> <p>190'</p>	

Unterrichtsplan

Walter Rothlin

9. Abend	REST Service komponieren aus bestehende- REST Services	<u>Gebäudeautomation</u> Selecta-Automat steuern Rolladensteuerung anhand Wettervorhersagen PiPlates Shellys	Probieren, Vorma- chen, Nachma- chen mit theoreti- schen kurzen Ein- schüben	200'	
----------	---	---	--	------	--

Bemerkungen:

- Jeder Abend dauert 4 Lektionen.
- Der Unterrichtsplan kann bei Bedarf dem vorhandenen Wissen der Klasse angepasst werden.
- Die Studierenden lösen die Übungen auf ihren privaten Notebooks.
- Der Leistungsnachweis am 8.Aband ist in Einzelarbeit in der vorgegebenen Zeit zu erstellen